

# Swarm Communication by Evolutionary Algorithms

Neil Vaughan, *Member, IEEE*

Department of Computer Science,  
University of Chester, United Kingdom  
Royal Academy of Engineering, London, United Kingdom  
n.vaughan@chester.ac.uk

**Abstract**—This research has applied evolutionary algorithms to evolve swarm communication. Controllers were evolved for colonies of artificial simulated ants during a food foraging task which communicate using pheromone. Neuroevolution enables both weights and the topology of the artificial neural networks to be optimized for food foraging. The developed model results in evolution of ants which communicate using pheromone trails. The ants successfully collect and return food to the nest. The controller has evolved to adjust the strength of pheromone which provides a signal to guide the direction of other ants in the colony by hill climbing strategy. A single ANN controller for ant direction successfully evolved which exhibits many separate skills including food search, pheromone following, food collection and retrieval to the nest.

**Keywords**—*ant colony; neuroevolution; pheromone communication; swarm communication.*

## I. INTRODUCTION

This research has developed a model of ant colony swarm intelligence behaviour. The novel aspect is that behaviour of pheromone navigation was not hard coded, as in most implementations, but has evolved using ANNs and an implementation of neuroevolution. Compared to previous research which failed to evolve standard and fixed topology ANNs for ant behaviour (Collins & Jefferson, 1990a), this research produces successful evolution and applies a more comprehensive neuroevolution methodology including complexification and augmentation of ANN topology and weights, as described by NEAT (Stanley, 2004).

Inspired by biological ants, this research aims to provide insights to advance understanding of how pheromone communication evolved in biological organisms. Application of neuroevolutionary computational modelling provides a useful analogy to how brains may have evolved to produce biological organism behaviours.

There are many long standing open questions regarding the evolution of altruism, related to how any why the evolution of cooperation emerged among closely related individuals (Hamilton 1964). Worker ants (Formicidae) are a perfect example of altruism, as they collect food for the good of the swarm but they get no individual rewards. This computer simulation method can provide new insights into altruism because each colony is only assessed by its fitness as a whole, not that of individual ants.

Pheromone trails can be seen as social memory or swarm memory used by all agents in the colony. The problem is called central place food foraging which is an optimisation problem. The aim of foraging is to collect as much food as possible and return it to the nest. An ant's food collection consists of two phases: the search for food and retrieval of food back to the nest.

In this respect the problem relates to the new field of morphogenetic engineering (ME). In this task, the core challenge posed by ME is a reverse engineering one: How can the ants' micro-rules be inferred from the system's macro-objectives? (Doursat et al., 2013). In this case the macro objective is to optimise fitness of the swarm by using swarm communication, but the micro rules for each ant to achieve that were not provided in this system and needed to evolve autonomously with no prefabricated design.

## II. RELATED WORK

Literature on pheromone communication is described by various key words: ant evolution simulation pheromone, central-place foraging algorithm (CPFA), pheromone recruitment (Letendre and Moses, 2013).

The core interest of this work is how ant pheromone communication can be evolved in a computational model. There have been some interesting works attempting to evolve ant pheromone communication, and others evolving swarm communication in general which is related closely enough.

A milestone early attempt to use a computer simulation to evolve ant foraging strategies using pheromones which resemble behaviours of biological ants was AntFarm (Collins & Jefferson, 1990a). AntFarm implemented an early form of neuroevolution, which was used to evolve the ANNs which learn behaviour for effective ant pheromone communication (Collins & Jefferson, 1990b). Neuroevolution methods in AntFarm evolved both the ANN connectivity pattern (topology) and weights of the ANN which were under genetic control in a genotype. Limitations were that: (1) AntFarm did not successfully evolve any cooperative foraging which was the main objective. (2) A basic, conventional ANN was used, when compared to the wider range of operators, sigmoids and activation functions with complexification as used in more recent neuroevolution models such as NEAT (Stanley, 2004). (3) The number of neurons and connections were not under genetic control.

The first research to evolve Ant pheromone foraging was by Panait and Luke (2004).

More recently, Beem (2017) attempted to use NEAT to evolve the controller for individual agents in a swarm. However the methods failed to produce any ability for agents to find food, or communicate, or exhibit any swarm intelligence whatsoever. The most advanced behaviour that his agents ended by evolving was to walk in circles. Perhaps that was due to the coordinate system used, or a lack of random or sin wave inputs. The inputs to the NEAT ant controller included the ant's own position; the intensity of pheromones at its location; whether or not the ant is carrying food at a given moment and the distance to the nearest food from two different points on the ant (for triangulation). The NEAT outputs are the agent's forward movement speed, its steering direction and the intensity of the pheromones it leaves behind. All agents within a swarm have the same neural network as controller.

Yong and Miikkulainen (2009) found that for cooperative tasks such as chase and evade, evolving several autonomous, cooperating neural networks to control a team of agents is more efficient and robust than evolving a single centralized controller. This potentially may apply in ants where two distinct roles are required – searching for food and returning to the nest.

Other attempts to evolve Swarm Intelligence using NEAT have failed, for example Chang & Worlanyo (2015) didn't see any communication being evolved. In other work, to some extent evolving swarm communication has succeeded (Floreano et al., 2007, Marocco and Nolfi., 2003, Yong and Miikkulainen., 2009). Rawal et al. (2012) successfully evolved cooperative communication between a group of predators who can only catch prey by communicating information codes to each other. A related work has evolved ants nest site localisation (Marshall, 2003).

Ant algorithms are generally most widely known through the wide literature on optimisation problems with ant colony (ACO) by Dorigo et al., (2006). Differing from this research, ACO algorithms conventionally must be hard coded by a designer and not evolved automatically. Other research applied ACO to adjust the ANN weights during neuroevolution (Conforth and Meng, 2008).

Letendre and Moses (2013) used genetic algorithms to show that ant foraging is improved in random food distributions and using both pheromone and site fidelity foraging strategies. However their GAs were used only to adjust a set of parameters affecting behaviour, not to learn the behaviours themselves, which were hard-coded and pre-existing.

### III. STATE OF OUTSTANDING PROBLEMS

There are some state of art current outstanding problems specifically within the evolving pheromone communication, some of which are addressed in this work.

Collins & Jefferson (1990a) suggested future work should involve: (1) a systematic study of the effect of food distribution

on the evolution of foraging strategies, testing the model of Johnson et al. (1987). (2) evolution of foraging strategies that are strongly affected by competition to see if competitor colonies sharing a single environment will interfere with each others strategies, disrupting communication by overwriting pheromone or misleading trails – which is related to Anti-pheromone which was later separately used by Panait and Luke (2004). (3) Investigate previous suggestions that pheromone evolution requires incremental changes to vary the environment, slowly making foraging more difficult over time.

Future work can also focus on the limitations of Panait and Luke (2004) which was suggested as future work. (1) When using multiple food sources which decay when eaten, this results in a dynamic changing environment and this makes pheromone evaporation more important. (2) How does pheromone navigation change with introduction of predators. (3) Future work can investigate ants which can produce more than 2 pheromones, so the ants can also learn complex tours with multiple way-points and self-intersecting paths.

### IV. RELATED WORK

There are five components (A-E) within the system which occur when an ant makes a move.

#### A. Pre-computed Inputs (ant sensors).

Ants have 13 input sensors (Described in Table 1 and Fig. 6): (1, 2) the location within the 9 adjacent cells (Moore neighbourhood) of the highest pheromone, (3, 4) the location within Moore neighbourhood which is closest to the nest, given by a 'compass sensor'. (5, 6) location within Moore neighbourhood of food. (7, 8) the direction of the ant's previous move, (9, 10) a direction picked at random, (11) a Boolean indicating whether the ant is currently carrying food, (12) a random number, (13) a fixed value of 1 (Bias). These are referred to as the pre-computed inputs and they remain the same even when the controller is changed (BM1, ANNs, NEAT).

Having a compass avoids the requirement to use two different pheromones. Compass is calculated by Pythagoras theorem using the x and y differences between ant and nest. In a grid system following the compass does not produce a direct path, it results in diagonal motion followed by perpendicular motion.

All of the 5 direction pre-computed inputs are represented in a consistent manner using two variables for x and y. These represent the change required in the ant's current x and y coordinates. These variables can be positive, negative or neutral. If both are neutral the ant would stand still (which would never be beneficial when foraging). If both were negative, the ant would move diagonally towards the origin (NW). With this method, the two variables can encode any direction within the 9 squares of the ant's Moore neighbourhood. If the ant chooses to follow the compass, it would then ignore the pheromone and vice versa.

### B. Controller.

The controller is a ‘black box’ brain which decides the animal behaviour at timestep  $t$ , based on the pre-computed inputs from the ant’s sensors. The experiments were repeated using different controllers: a hard-coded benchmark (BM1), a fixed topology neural network and neuroevolution by adjusting the topology and weights of an ANN.

### C. Outputs.

The resulting output of the controller determines the direction in which the ant moves.

### D. Post move local updates.

After each ant has moved, a number of post-move local updates are automatically applied. (1) If the ant is now standing on food and isn’t carrying any, it automatically picks food up. (2) If the ant is carrying food, pheromone is deposited with strength inversely proportional to the time since collection. (3) If an ant is already carrying food and is now standing on a nest, it automatically drops the food. This representation realistically assumes that biological ants already could pick up and drop food before they evolved pheromone communication. These tasks are regarded as automatic responses which we assume have been learnt previously.

### E. Global updates.

After a full iteration, when all ants have finished making a move, a global update is triggered in which all pheromone is evaporated (decremented). A number of different evaporation rates including decrementing and various percentage reductions were tested to identify how evaporation rate affects the ability to evolve navigation controllers.

## V. THE MACHINE LEARNING TASK

The given inputs and expected outputs were kept strictly equal for all tested controllers. Therefore here we can formally define the machine learning task based on the relationship between inputs and outputs of the controller. This is critical step because small changes to the representations of input and output can make big changes to the difficulty of the task for the controller to learn.

In total the task has 13 inputs: 10 (5 pairs of) input direction variables, 1 boolean input, 1 random number input and 1 fixed value (Bias). There are 2 outputs:  $x$  and  $y$ .

Of the 13 inputs 10 inputs are positional change inputs. These are in 5 pairs of  $x$  and  $y$ , relative to ants current position, to reach the optimal square within Moore neighbourhood for the 5 pre-computed inputs: food, pheromone, compass, same-move or random-move. These all have three possible values:  $-1, 0, 1$ .

For food and pheromone,  $0, 0$  only occurs when none is found, which means that there is no need for having separate Booleans indicating food and pheromone presence. For compass,  $0, 0$  only occurs when standing on the nest (in which case compass would not be useful as the ant would not be carrying food because it would have been dropped automatically.

There is one random number input called  $r$ . This is independent of random direction inputs. This is required so that ants can randomly determine when to move randomly.

A Boolean is included to represent whether food is currently being carried. This is an important flag because it defines one of two current states: (1) searching for food, or (2) bringing food back to nest. This information is not available in other inputs.

There are only two outputs. They represent the relative step the ant will take on this timestep. They can be a value from the set  $\{-1, 0, 1\}$ . Therefore the output of the controller purely determines the position of the ant’s next move, which has 32 options, one for each square in the ant’s Moore neighbourhood.

TABLE I. OVERVIEW OF ALL INPUTS AND OUTPUTS TO THE EVOLVING NEURAL NETWORK

Input/Output Name	Description
Closestnestx, Closestnesty	The location which is closest to the nest within Moore neighbourhood (compass sensor). $(-1;0;1, -1;0;1)$
Foodherex, Foodherey	The location of food in the Moore neighbourhood, $(-1;0;1, -1;0;1)$ format and $(0,0)$ if not present.
Highestpheromonex, Highestpheromoney	The location of the highest pheromone in the Moore neighbourhood, in format $(-1;0;1, -1;0;1)$ . If not present $(0,0)$ .
rand_x, rand_y	A randomly selected location in the Moore neighbourhood. $(-1;0;1, -1;0;1)$
same_x, same_y	The location within the Moore neighbourhood which continues the previous direction (opposite the ant’s previous location). $(-1;0;1, -1;0;1)$
Carryingfood	A Boolean flag: 1 (True) if carrying food; 0 (False) otherwise.
$r$	Independent random number input (floating point $0-1$ )
output_x, output_y	The ANN output represents the $x$ and $y$ to move into $(-1;0;1, -1;0;1)$ .

## VI. BENCHMARK ALGORITHM BM1

The developed system included designing a custom developed hard-coded benchmark algorithm (BM1) for pheromone based food foraging behaviour, shown as pseudocode in Fig. 1. The BM1 algorithm was used in this research as a comparison or gold standard to assess the performance of the fixed topology and evolving ANN algorithms.

```

if (carryingfood){
    //follow compass to nest.
}else if (food in neighbourhood){
    //step onto the food.
}else if (pheromone in neighbourhood){
    //step onto strongest pheromone.
}else if (rand%100<90){
    //continue previous direction.
}else{
    //use a completely random direction.
}

```

Fig. 1. Pseudocode for the Benchmark Algorithm (BM1).

The benchmark BM1 does produce efficient foraging behaviour and also demonstrates that the pre-computed inputs provide all required information to complete the foraging task. The benchmark performance was measured and used to evaluate the performance of ANN driven behaviour controllers which later evolved. The pseudocode gives a description of what happens for each ant to decide which direction to move in at each timestep. This implements two modes: searching for food, and retrieving food based on the carryingfood Boolean flag.

In the BM1 benchmark, when a random direction has been chosen, on consecutive timesteps, the direction has 90% chance of remaining constant. This means that ants travel largely in straight lines, broken by abrupt changes on 10% of steps. This causes the ants to more effectively spread out and cover the whole grid more quickly. The main benefit is that ants then have a much higher chance of running into an existing pheromone trail. This outperforms total random movement, whereby ants often retrace their steps in consecutive turns which results in a lack of general directed movements. Also the random direction is chosen from an 8-square Moore neighbourhood – so that standing still is never chosen as it would have no benefit.

It can be seen that efficient pheromone communication (BM1) can be captured in this simple pseudocode which consists of only 5 IF statements, plus the defined actions to perform within each condition. The machine learning task is to replicate the behaviour of these 5 IF statements and associate the correct actions with each case, by using only the 12 given inputs. This summarises the difficulty of the learning task. If the machine learning fails, it must be because the IF statement structure was too complex to learn, or the actions were not associated with the correct conditions. The BM1 already demonstrates that the given 12 inputs are satisfactorily informative to complete the foraging task.

In the event of machine learning failure, that could be investigated in terms of machine learning complexity, rather than anything specifically about the foraging task, because it could be assumed that other machine learning tasks with the same level of complexity would equally fail to be learnt.

## VII. RESULTS OF BM1 IN FIXED LEVELS

For a comparison between controllers (BM1, ANN, Neuroevolution), one fixed level was used. The obstacles and food were located in the same places. That ensured that each controller was subject to the exact same challenge. The fixed level is shown in Fig. 2.

When the ant colony was controlled by the BM1 algorithm and foraged within this fixed layout level (Fig. 2.), the ant colony makes very consistent progress every time it is run (Fig. 3.). The small variations are due to the random movement of ants, taking slightly different times to first discover food sources before they are subsequently attended by large recruited swarms.

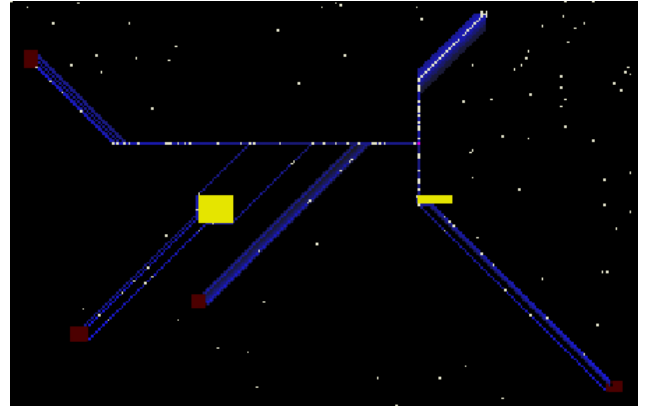


Fig. 2. This shows the layout of the fixed level used for evaluating and comparing a variety of controllers.

In total the fixed level happens to have 6630 foods. On a typical run in this fixed level, as those shown on the graph Fig. 3, BM1 collected the first food after 106 timesteps. At 2500 timesteps, 3143 foods were collected. By the time the run was halted at the 5000<sup>th</sup> timestep, food was still being actively collected, in total 4852 foods had been collected, so 1778 foods remained uncollected.

The behaviour of the BM1 can be further analysed by looking at Fig 4. which shows how often each direction was chosen. Standing still is the rarest move and diagonal bottom-right to top-right the most frequent. Also Fig. 5. shows how often each of the 5 IF statements from the pseudocode (Fig. 1) were triggered. Continuing in the same direction is the most frequent action and stepping onto food is the rarest action.

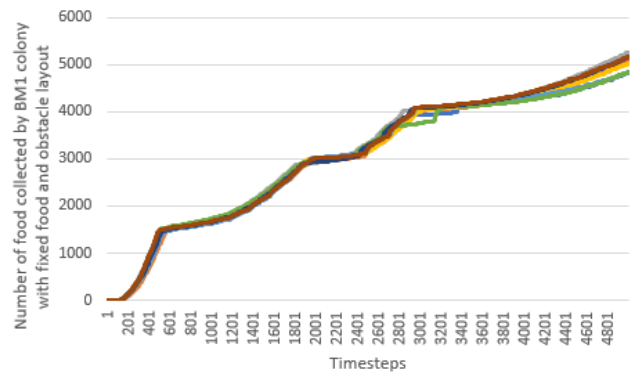


Fig. 3. Consistent results of BM1 run 8 times on a fixed level.

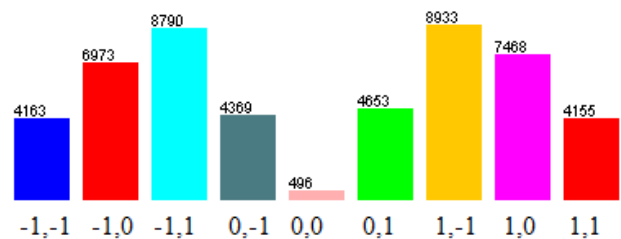


Fig. 4 How often each Moore neighbourhood direction was chosen.

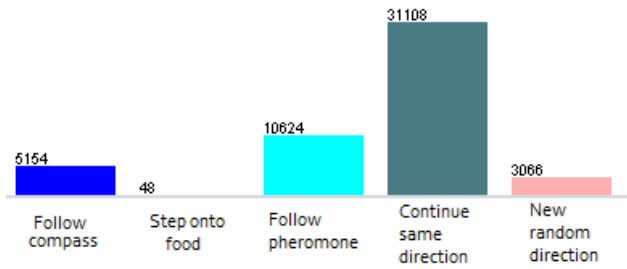


Fig. 5. How many times BM1 triggered each IF statement from the pseudocode.

## VIII. FIXED NEURAL NETWORK TRAINING

Using a typical run of BM1 in a fixed level, as in Fig. 1. a training set for an ANN was produced. This was achieved by writing to file all of the inputs and the resulting outputs for 10 ants over a 5000 timestep run. This data set contained 50,000 instances each containing a full set of 13 inputs and 2 outputs.

The training set was then used as the training set for a fully connected static neural network. The aim was to identify how accurately backpropagation would perform to train a NN to learn the relationships between the inputs and the output produced by BM1. A 90% split was used to split into a 45,000 instance training set and an unseen 5,000 instance test data.

In order to use a single ANN to produce two outputs: x and y together, output\_x and output\_y were combined into a single output class with 9 values A-I (Fig 6). In total the ANN had 12 inputs and 9 output nodes, one for each class. With no hidden layers, the network classified 87.5% correctly. With one hidden layer of 10 nodes the MLP correctly classified 90.4% of test data. With two hidden layers of 10 nodes each (Fig. 6), accuracy improved to 92.3%. Training time increased with extra hidden layers.

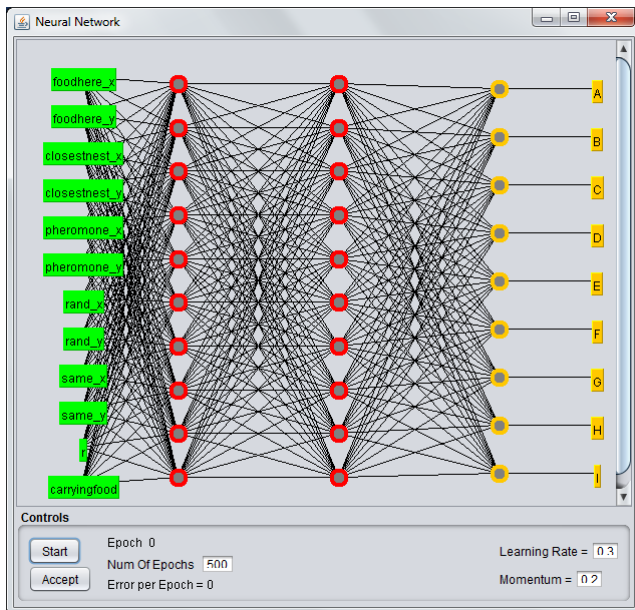


Fig. 6. The fixed topology ANN classified the optimal ant direction 92.3% of the time.

Classification accuracy is defined as how often the fixed ANN produced the same movement decision as the BM1 algorithm.

It is not known if this trained ANN would in fact perform well as a controller for the ant simulation, or not. The misclassified instances could include important classes. It is hard to identify which situations the network failed in and if those would be critical or not to foraging behaviour.

To summarise this fixed ANN training section, in this approach BM1 output data was used to train an ANN by back propagation. This step was useful to demonstrate that an ANN is capable of performing this task when properly trained, so therefore the task should not be overly complex for an evolving neural network.

This result suggests that a trained fixed topology ANN can perfectly learn to recognise which of the 5 IF statements should be triggered given the inputs. A harder learning task is to also determine which actions to take when each of the 5 IF statements are triggered, the ANN achieved this with 92.3% accuracy.

It is recognised that this approach differs from biological ants with natural selection, which have no pre-existing data to train the ants. In evolution, skills must be evolved without training or guidance, not towards a particular aim or objective. Therefore, the next section focusses on unsupervised evolution.

## IX. EVOLVING ANN CONTROLLERS

Neuroevolution was applied to evolve neural networks which were then applied as the controller for ants. All ants in a colony had the same controller at each generation. But between generations, the controller was subject to genetic change, by modifying the ANN both in terms of the weights and the topological structure, including the number and location of connections.

Initially, the ANNs were set blank, with no hidden layer nodes. The additional nodes are added by evolution over time.

The fitness function was defined as 1 point for each food picked up and 50 points for each food returned to the nest.

A comprehensive set of tests was done with 25 ants per colony, left to run for 900 timesteps. The ANN controllers were subject to neuroevolution in populations of size 15 organisms over 100 generations and this was repeated 10 times. Afterwards a further 5 repetitions of 100 generations was completed, this time with populations of 150 organisms which is a more conventional population size for neuroevolutionary algorithms.

In all runs food collection was learnt almost perfectly. By the 10<sup>th</sup> generation organisms often had peaked at a fitness of 25, meaning that every single ant successfully collected a food. In most experiments, the ANNs started to learn to return the food to the nest, which begun producing fitness of 62 in generation 13. At generation 42 the fitness was 2049, so the majority of ants were returning multiple foods to the nest. This cannot be explained by random movement alone which does not result in any food being returned. In the third run, the



highest fitness reached 5059, meaning that 109 foods were collected, of which 99 foods were returned to the nest in only 900 timesteps, a highly efficient result, that means every ant on average collected food and returned it to the nest 4 times, outperforming BM1.

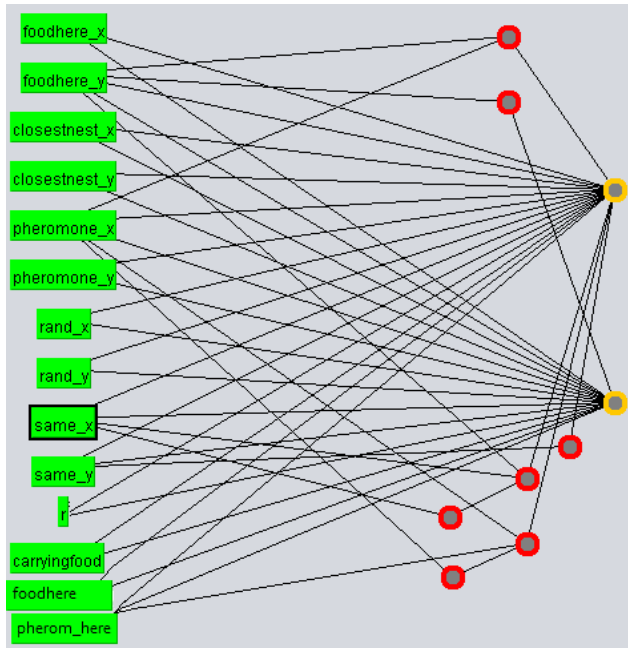


Fig. 7. The evolved ANN which has fitness 5059 and outperformed the benchmark algorithm. Red nodes were added by the neuroevolutionary algorithm, the two yellow nodes are outputs, one for the x movement, one for y movement. The green squares are the 14 ANN inputs.

Visualisation of the evolved ANN structure (Fig. 7) shows that it had an additional 7 nodes had evolved, and 18 new connections, plus all of the weights throughout had evolved to optimal values.

An experiment used two additional Boolean inputs flags – foodhere and pherom\_here, to indicate whether food or pheromone were located in the neighbourhood. These flags were not necessary and evolution could work just as well with only the 13 inputs.

An experiment was conducted using the same inputs, but with a hidden layer of 10 hidden nodes, fully connected. This could have been useful if the learning is too complex to be evolved using inputs directly connected to outputs. For example a Boolean input flag carryingfood needs to cause the controller to switch between either following the compass or the search strategy. This logic may not be possible to represent in a network with no hidden nodes. Eventually neuroevolution should add a node then enables this logic to be modelled, however, with some hidden nodes to start with, all that should be required is to adjust the weights only and not the topology. It was thought that this could enable the ideal behaviour to be learnt faster.

The hidden layer resulted in extra burden of calculating weights for over 140 links compared to only 30 in previous runs. Results show that a hidden layer didn't make any noticeable decreases to learning speed or increased

performance. Perhaps this is because 10 hidden nodes was more than necessary, or because neuroevolution may find it just as challenging to adjust all the extra weights in the increased search space as the topology.

Running the BM1 on random levels produced fitness of 49300 which equates to 98.6 foods returned to the nest per level. Fitnesses of the evolved controller reached over 2/3 of this score.

## X. EVOLVED SEARCH STRATEGIES FOR RANDOM LAYOUTS

To evolve a general-purpose foraging algorithm, several experiments were completed using random level layouts. Previously, fixed levels were used to compare neuroevolution to BM1 (neuroevolution outperformed the hard-coded benchmark), but ants evolved to always simply search for food in the direction of the nearest food, because food was always in the same place. The random level stops that strategy from working. In random layouts no food was ever placed closer than 50 squares from the nest, however that doesn't stop some levels from being easier than others. The difference in level difficulty should average out by natural selection over long runs. But variable difficulty of the layouts does create a difficulty for assessing fitness fairly. To reduce this effect each organism was run on ten different random levels each. Results show that different search strategies evolved in random levels that are more general purpose.

One evolved strategy is using a well spaced random walk (Fig. 8). Compared to the previous evolved strategy of heading in only one direction, it can find food in a random level far more often.



Fig. 8. An evolved search strategy: well spaced random walk.

Secondly a line scanning method evolved in which all agents spread along a vertical line so that any food over a wide area covering around 20 degrees is effectively located (Fig. 9). The line scanning approach may demonstrate working as a group as this approach could be less effective with individual agents. This strategy seemed the most effective evolved solution and it evolved several times independently.

Some of the evolved strategies had clearly exploited properties of the task design. Rather than searching a wide area as expected, several strategies move in a tightly clustered group. By covering a smaller area, chances of finding food were reduced, but that is cancelled out when a food is found and all ants find it together and return a much larger amount.

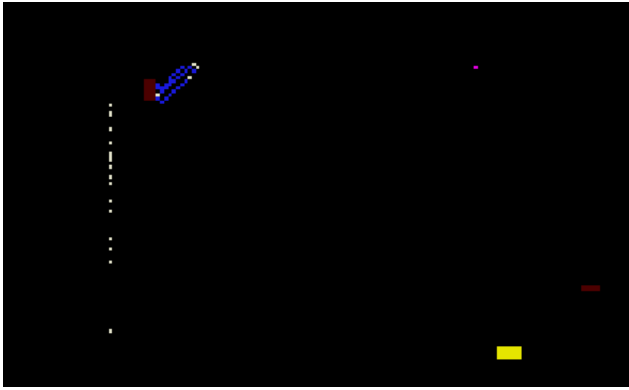


Fig. 9. An evolved search strategy: line scanning.

## XI. LEARNING PHEROMONE DEPOSIT STRENGTH

This test identified whether a single ant controller could learn the appropriate pheromone strength to deposit in addition to which direction to move.

In ideal ant foraging behaviour, the strength of pheromone deposits is strongest when food has recently been picked up and decreases proportionately with distance travelled from the food source. Biological ants must have evolved this skill because it enabled other ants to find the food using a hill climbing strategy. Critically, pheromone strength reduction must occur faster than pheromone evaporation rate, otherwise hill climbing would lead away from the food. An experiment was conducted to see whether the neuroevolution controller could learn to appropriately control the pheromone deposit strength whilst simultaneously controlling the direction of travel. This required an additional input to the ANN, describing the number of steps taken since food was picked up. Also one additional output was added which signals the strength of pheromone to deposit. During early evolutionary phase in the initial runs before learning had taken place, pheromone was deposited almost randomly as the ants moved randomly (Fig. 10).

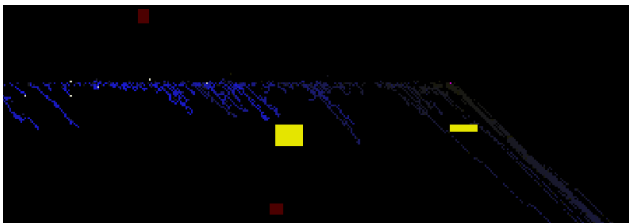


Fig. 10. Random deposits result from controlling the pheromone strength with an ANN in early evolution.

After more evolution, a new strategy emerged, which was to lay pheromone only when searching for food (Fig. 11). This was the opposite of what was expected but pheromone could be used to find a way back to the nest along with compass.

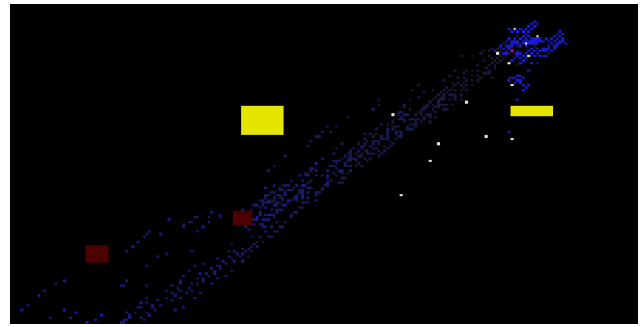


Fig. 11 Evolving a single ANN for direction controller and the pheromone strength. Ants learnt to bring food back to nest but pheromone is not correctly used. This was after 15 population size.

Other strategies evolved which resulted in complex structures (Fig. 12).

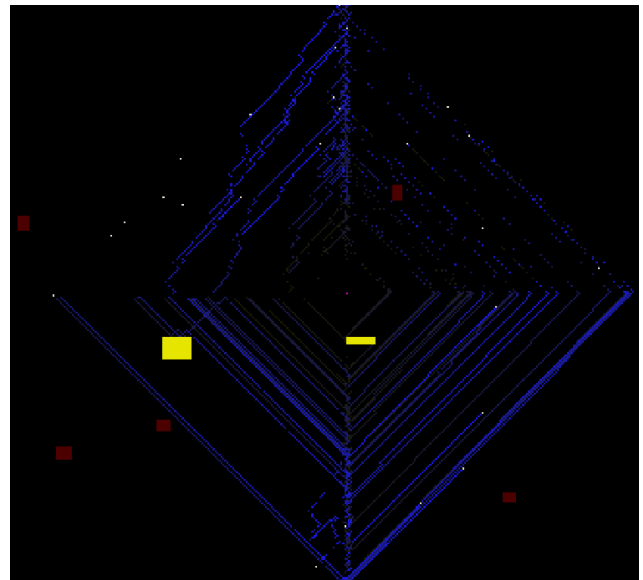


Fig. 12. An evolved strategy leaves complex patterns of pheromone trails.

In this situation none of the many evolved controllers ever appeared to be controlling both direction and pheromone optimally together. Future experiments could further investigate whether that was because of the limited population size of 15. The ants were bringing food to the nest as individuals but not communicating

A common problem when evolving pheromone is that normally when an ant is laying pheromone the ant should not follow pheromone at the same time and conversely when following pheromone an ant should not lay pheromone. If both are done at the same time, an ant can follow itself. This became a serious problem in the tests when all ants started moving around together in one group following the pheromone trails that other ants were leaving which led nowhere. To avoid that, it was necessary that all ants take one step and then afterwards all ants leave pheromone. Ultimately the ants need to stop pheromone when not carrying food otherwise problems arise.

## XII. LEARNING PHEROMONE STRENGTH ALONE

To simplify the machine learning task, it was split into two tasks. It is commonly observed that when two distinct tasks are to be evolved, two ANNs are better rather than one trying to learn two different skills. In this task there is movement, and decision about what strength pheromone to leave. If pheromone decision is to be learnt this is a much simpler task, alone, with only 2 inputs (carryingfood and stepssincefood) and one output to control pheromone deposit strength. This pheromone decision was implemented using neuroevolution combined with movement direction controlled by BM1.

In that case there was quick success evolving pheromone strengths. Initially, some ants evolved to leave pheromone only when carrying food which was good behaviour, but they were not adjusting the pheromone strength when getting further from the food source. These trails led other ants in the wrong direction getting trapped at the nest, so these colonies didn't score a very high fitness (Fig. 13). With the 3 input 1 output network, many of the first organisms laid only fixed strength pheromone so that hill climbing sent ants into a one way trap in the opposite direction.

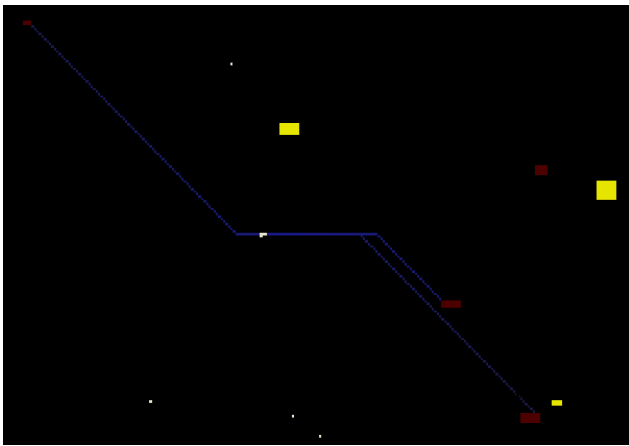


Fig. 13. Evolved pheromone trails leading in the wrong direction.

Other evolved controllers left local optimums in the pheromone trail so hill climbing didn't work (Fig. 14). Various ways to convert stepssince into pheromone strength evolved, of which only few were beneficial. Without any communication these colonies collected only 5-10 foods in each run.

Later after evolution, controllers evolved which had successfully adjusted pheromone strength for hill climbing. With effective communication colonies collected over 50 foods in a single run. This was more of a trivial task for neuroevolution to solve and even a correct solution was found in generation 1. However many generations later the correct behaviour had not propagated into all organisms because the good solution was mutated even after the correct solution was found.

For learning pheromone it doesn't matter that fixed levels or random levels are used, because the search for food is not being tested or evolved at this stage.

In this task fitness was effectively Boolean related to whether the pheromone was actually compatible with hill climbing or not.

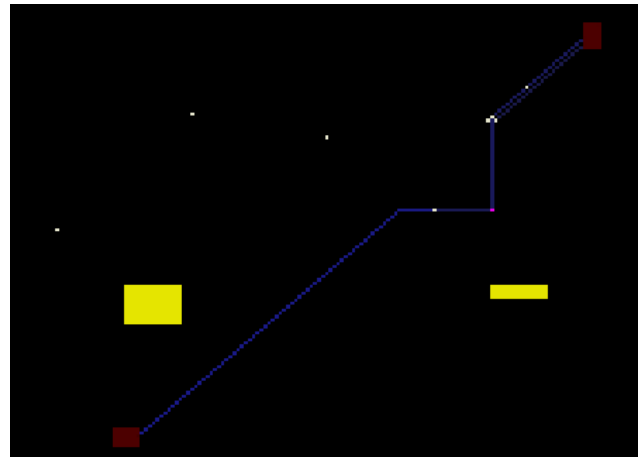


Fig. 14. Local optimums in the pheromone strength cause hill climbing strategy to fail. But neuroevolution quickly fixed that.

## XIII. CONCLUSIONS & FUTURE WORK

This paper has investigated the benefits of neuroevolution compared to fixed topology ANN by testing how pheromone behaviour can evolve in both, in relation to a hard coded designed benchmark (BM1).

This research applied evolutionary algorithms to evolve swarm communication. Controllers were evolved for colonies of artificial simulated ants during a food foraging task which communicate using pheromone. Neuroevolution enables both weights and the topology of the artificial neural networks to be optimized for food foraging. The developed model results in evolution of ants which communicate using pheromone trails. The ants successfully collect and return food to the nest. The controller has evolved to adjust the strength of pheromone which provides a signal to guide the direction of other ants in the colony by hill climbing strategy. A single ANN controller for ant direction successfully evolved which exhibits many separate skills including food search, pheromone following, food collection and retrieval to the nest.

Future work could investigate simulating other colony or swarm intelligences with communication. Examples include smells in the air or environment such as territory marking, or sounds used for predator detection, warnings or communication.

This paper has demonstrated neuroevolution applied to evolve pheromone communication in simulated ant colonies. The core intelligence required to perform pheromone communication was summarised in form of the hard coded benchmark BM1, comprised of an IF block with 5 conditions. The 5 conditions were learnt by a fixed ANN and the actions to take within each IF statement were learnt with 92% accuracy.

Differences between neuroevolution, fixed ANN and BM1 were demonstrated. BM1 has disadvantages that it has to be programmed and designed which could be challenging for more complex tasks and the optimal method may not be



implemented. A fixed ANN could replicate the behaviour of BM1 with 92% accuracy. Neuroevolution was able to learn the tasks required to follow pheromone and separately to control strength of deposited pheromone.

#### ACKNOWLEDGMENT

The research was funded by The Royal Academy of Engineering (RAEng) as part of the Research Fellowship awarded to Neil Vaughan as Principle Investigator.

#### REFERENCES

- [1] Gaggioli. Optimal Experience in Ambient Intelligence. In: Ambient Intelligence. G. Riva, F. Vatalaro, F. Davide, M. Alcáñiz (Eds.). IOS Press (2005).
- [2] Jin, A., & Austin, M. (2014). Comparative Evolutions of Swarm Communication. Swathmore College. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.486.9490>
- [3] Marocco D, Nolfi S. (2003) "The Emergence of Communication in Evolutionary Robots". In: Philosophical Transactions of the Royal Society London A.361, pp. 2397–2421.
- [4] Baddeley, B., Graham, P., Husbands, P., & Philippides, A. (2012). A model of ant route navigation driven by scene familiarity. PLoS computational biology, 8(1), e1002336.
- [5] University Of Sussex, (2006), How ants find their way, Science Daily, <https://www.sciencedaily.com/releases/2006/10/061018094651.htm>
- [6] Collins, R. J., & Jefferson, D. (1990a). Antfarm: Towards simulated evolution. Artificial Life II Conference, Santa Fe, NM. In J. D. Farmer, C. Langton, S. Rasmussen, and C. Taylor (Eds.), Artificial Life II, Addison Wesley, 1991.
- [7] Marshall, J., Kovacs, T., Dornhaus, A., & Franks, N. (2003). Simulating the evolution of ant behaviour in evaluating nest sites. Advances in Artificial Life, 643-650.
- [8] Hamilton, W. D. (1964). The genetical evolution of social behaviour. II. Journal of theoretical biology, 7(1), 17-52.
- [9] Adler, F. R., & Gordon, d. M. (2003). Optimization, conflict, and nonoverlapping foraging ranges in ants. The American Naturalist, 162(5), 529–543.
- [10] Doursat, R., Sayama, H., & Michel, O. (2013). A review of morphogenetic engineering. Natural Computing, 12(4), 517-535.
- [11] Wohlgenuth, S., Ronacher, B., & Wehner, R. (2001). Ant odometry in the third dimension. Nature, 411(6839), 795–798.
- [12] Hölldobler, B. (1976). Recruitment behavior, home range orientation and territoriality in harvester ants, Pogonomyrmex. Behavioral Ecology and Sociobiology, 1(1), 3–44.
- [13] Thielin-Bescond, M., & Beugnon, G. (2005). Vision-independent odometry in the ant Cataglyphis cursor. Naturwissenschaften, 92(4), 193–197.
- [14] Beem FV, 2017, Analysis of NEAT and application in swarm intelligence, Amsterdam University, Undergraduate Dissertation, Supervisor: Rein van den Boomgaard.
- [15] Chang R, Worlanyo S, 2015, Evolving Swarm Communication with NEAT, Swathmore College Project Report, <https://pdfs.semanticscholar.org/09ee/9844e87c3fa7571a934efbf81efc36172802.pdf>
- [16] Conforth and Meng, (2008) Toward Evolving Neural Networks using Bio-Inspired Algorithms. In IC-AI, pp. 413-419.
- [17] Rawal, A., Rajagopalan, P., Miikkulainen, R., & Holecamp, K. (2012). Evolution of a communication code in cooperative tasks. Artificial Life, 13, 243-250.
- [18] Collins, R. J., & Jefferson, D. R. (1990b). An artificial neural network representation for artificial organisms. In International Conference on Parallel Problem Solving from Nature (pp. 259-263). Springer, Berlin, Heidelberg.
- [19] Panait, L., & Luke, S. (2004). Learning ant foraging behaviors. In proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9) (pp. 575-580).
- [20] Letendre K and Moses ME (2013). Synergy in ant foraging strategies: Memory and communication alone and in combination. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO '13 Companion), pages 41–48, New York, NY. ACM.
- [21] T. Paz Flanagan, K. Letendre, W. Burnside, G. M. Fricke, and M. Moses. How ants turn information into food. In 2011 IEEE Symposium on Artificial Life (ALIFE 2011), pages 178–185, Piscataway, NJ, 2011. IEEE Press.
- [22] Hecker, J. P. (2015). Evolving Efficient Foraging Behavior in Biologically-Inspired Robot Swarms (Doctoral dissertation, The University of New Mexico).
- [23] Stanley, K. O. (2004). Efficient evolution of neural networks through complexification (Doctoral dissertation).
- [24] Yong, C. H., & Miikkulainen, R. (2009). Coevolution of role-based cooperation in multiagent systems. IEEE Transactions on Autonomous Mental Development, 1(3), 170-186.
- [25] Floreano, D., Mitri, S., Magnenat, S., & Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. Current biology, 17(6), 514-519.
- [26] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.
- [27] Johnson, L. K., Hubbell, S. P., & Feener Jr, D. H. (1987). Defense of food supply by eusocial colonies. American Zoologist, 27(2), 347-358.